

Express Mailing Label No. EL487543635US

PATENT APPLICATION

Docket No. 3000.2.38

IBM No. ST9-99-145

UNITED STATES PATENT APPLICATION

of

SHYH-MEI HO

and

HOLGER JUSCHKEWITZ

for

REPRESENTING IMS TRANSACTION DEFINITIONS AS XML
DOCUMENTS

MADSON & METCALF, P.C.

ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Relevant Technology

With the explosive growth of the Internet, most of the world's computer systems are now interconnected or capable of being interconnected. However, in order to share data, the systems need to understand each other's data formats. In recent years, the computer industry has evolved at such a rapid pace that systems developed only a few years apart use vastly different and incompatible formats. Such incompatibility problems tend to increase with the "age" differences of the systems.

The Information Management System (IMS) is one of the oldest and perhaps the most popular transaction processing (TP) systems. A TP system supervises the sharing of resources for concurrently processing multiple transactions, such as queries to a database. Anyone who has ever used an ATM, rented a car, or booked a flight, has probably used IMS.

IMS was developed by IBM the 1960's as a inventory tracking system for the U.S. moon landing effort. Today, interfacing IMS with newer systems, particularly with systems of different manufactures over the Internet, is problematic.

As illustrated in Figure 1, an IMS typically includes two major components: an IMS Transaction Monitor (IMS/TM) 12, which is responsible for scheduling, authorization, presentation services and operational functions, and a hierarchical database 14, DL/I. Both components are independently configurable. For example the IMS/TM 12 can use a relational database, such as DB/2, rather than

1 DL/I. The various components of an IMS 10 communicate via the MVS operating
2 system 16.

3 As illustrated Figure 2, the architecture of IMS is divided into four regions. For
4 example, a Message Processing Region (MPR) 20 is used to execute message-driven
5 applications 18. Execution of applications 18 in this region 20 is triggered by
6 incoming messages, such as those received from a terminal.

7 By contrast, applications 18 in a Batch Message Processing (BMP) 22 region
8 are not message driven. They are usually scheduled to run at times of low system
9 activity, such as nights and weekends. Typically, such applications 18 perform a
10 number of predefined operations, after which they immediately terminate.

11 An Interactive Fast Path (IFP) 24 region allows fast and simple requests to the
12 database 14. Applications 18 operating in the IFP 24 bypass the normal scheduling
13 mechanism, providing relatively fast response times. In general, IFP applications
14 18 stay resident even if they are not needed.

15 An IMS Control Region (IMSCTL) 26 is responsible for all TP tasks, as well
16 as for controlling all dependent regions (MPR 20, BMP 22, and IFP 24). Essentially,
17 the IMS Control Region 26 has three main responsibilities: telecommunications,
18 message scheduling, and logging/restart.

19 For example, IMSCTL 26 controls the connected terminals 28 (illustrated in
20 Figure 3), receiving/sending messages from/to the terminals 28. Moreover,
21

1 IMSCTL 26 logs all transactions in order to provide the capability of undoing non-
2 committed transactions in the event of a system failure.

3 In addition, every time IMSCTL 26 receives a message from a terminal 28, it
4 schedules an application 18 to process the message. IMSCTL 26 identifies the
5 desired application 18 and puts the message in the application's message queue 30.
6 The application 18 processes the requests in its message queue 30 and responds to
7 the originating terminal 28 by placing the response in the terminal's message queue
8 32.

9 As illustrated in Figure 4, an IMS 10 obtains all of its information about the
10 structure and behavior of its components (applications, databases, transactions, etc.)
11 from macro statements 34 (hereinafter "macros"). Certain macros 34 are referred
12 to as "transaction definitions" 35 because they define how transactions are
13 processed.

14 For example, as shown in Figure 4, an application (APPLCTN) macro 36
15 defines the behavior of a particular IMS application 18. An APPLCTN macro 36
16 exists for each application 18 in the IMS 10, and defines, for example, the
17 application's name, resource requirements, and appearance.

18 An APPLCTN macro 36 is followed by a zero or more (TRANSACT) macros
19 38, which define the various transactions applicable to the application 18. A
20 TRANSACT macro 38 specifies the appearance of a transaction to be performed by
21 an application 18, identifying whether the transaction is IMS exclusive, IMS Fast

1 Path potential or IMS Fast Path exclusive. Furthermore, a TRANSACT macro 38
2 specifies the transaction code that causes the application 18 named in the APPLCTN
3 macro 36 to be scheduled for execution in an IMS processing region.

4 Currently, IMS is only capable of processing transactions previously defined
5 by the APPLCNT and TRANSACT macros 36, 38. A user may not initiate arbitrary
6 transactions, such queries of the database 14, that have not been previously defined.
7 Moreover, in order to change the above-described macros, one must initiate a
8 process called "system generation," which necessitates shutting down the IMS 10
9 for a period of time.

10 In addition, the above-described IMS macros 36, 38 have a proprietary
11 format, which is a detriment in interfacing with remotely located systems from
12 different vendors. Currently, the dominant Internet format is the HyperText
13 Markup Language (HTML), a variant of the eXtensible Markup Language (XML).
14 Providing a technique for delivering IMS transactions definitions to an IMS 10 using
15 interchangeable documents, such as XML documents, would be a first step in being
16 able to initiate arbitrary IMS transactions over the Internet.

17 Accordingly, what is needed is a system and method for representing IMS
18 transaction definitions in an interchangeable format, such as XML. What is also
19 needed is a system and method for communicating with an IMS 10 using XML
20 documents. In addition, what is needed is a system and method for creating a
21

1 Document Type Definition (DTD) for use by a parser in converting between IMS
2 transaction definitions and XML documents.

SUMMARY OF THE INVENTION

The present invention solves many or all of the foregoing problems by providing a system and method for communicating with an Information Management System (IMS) using eXtensible Markup Language (XML) documents.

In one aspect of the invention, a method includes the steps of receiving a document comprising an IMS transaction definition encoded in XML; obtaining a Document Type Definition (DTD) specifying rules for decoding the IMS transaction definition; parsing the XML document using the DTD to decode the IMS transaction definition; and providing the decoded IMS transaction definition to the IMS.

In another aspect of the invention, a method includes the steps of modeling an IMS transaction definition in a Universal Modeling Language (UML) to produce a UML object model; and processing the UML object model using an XML Metadata Interchange (XMI) utility to create the DTD.

In yet another aspect of the invention, a method includes the steps of obtaining an IMS transaction definition; obtaining a Document Type Definition (DTD) specifying rules for encoding the IMS transaction definition; and parsing the IMS transaction definition with the DTD to encode the IMS transaction definition in an XML document.

In still another aspect, a system includes a document reception module configured to receive a document comprising an IMS transaction definition encoded in XML; a parser configured to obtain a Document Type Definition (DTD) specifying

These and other objects, features, and advantages of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is more fully disclosed in the following specification, with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of an Information Management System (IMS);

Figure 2 is a schematic block diagram of IMS processing regions;

Figure 3 is a schematic block diagram of message processing within an IMS;

Figure 4 is a schematic block diagram of IMS macros including two IMS transaction definitions;

Figure 5 is a schematic block diagram of a system for communicating with an IMS using eXtensible Markup Language (XML) documents according to an embodiment of the invention;

Figure 6 is a schematic block diagram of a system for generating XML documents from IMS transaction definitions according to an embodiment of the invention;;

Figure 7 is a schematic block diagram of a system for generating a parser, a number of access classes, and a Document Type Definition (DTD) for IMS transaction definitions according to an embodiment of the invention;

Figure 8 is a schematic block diagram of a Universal Modeling Language (UML) object model representing IMS transaction definitions according to an embodiment of the invention;

1 Figure 9 is a schematic flowchart of a method for communicating with an IMS
2 using XML documents according to an embodiment of the invention;

3 Figure 10 is a schematic block diagram showing a conversion between an
4 XML document and an IMS transaction definition according to an embodiment of
5 the invention;

6 Figure 11 is a schematic flowchart of a method for converting an IMS
7 transaction definition into an XML document according to an embodiment of the
8 invention;

9 Figure 12 is a schematic block diagram showing a conversion between an IMS
10 transaction definition into an XML document according to an embodiment of the
11 invention; and

12 Figure 13 is a schematic flowchart of a method for generating a parser, a
13 number of access classes, and a DTD for IMS transaction definitions according to an
14 embodiment of the invention.

1 **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

2 Certain presently preferred embodiments of the invention are now described
3 with reference to the Figures, where like reference numbers indicate identical or
4 functionally similar elements. The components of the present invention, as
5 generally described and illustrated in the Figures, may be implemented in a variety
6 of configurations. Thus, the following more detailed description of the
7 embodiments of the system and method of the present invention, as represented in
8 the Figures, is not intended to limit the scope of the invention, as claimed, but is
9 merely representative of presently preferred embodiments of the invention.

10 Throughout the following description, various system components are
11 referred to as "modules," "units," "utilities," "tools," or the like. In certain
12 embodiments, these components may be implemented as software, hardware,
13 firmware, or any combination thereof.

14 For example, as used herein, a module may include any type of computer
15 instruction or computer executable code located within a memory device and/or
16 transmitted as electronic signals over a system bus or network. An identified
17 module may include, for instance, one or more physical or logical blocks of
18 computer instructions, which may be embodied within one or more objects,
19 procedures, functions, or the like.

20 The identified modules need not be located physically together, but may
21 include disparate instructions stored at different memory locations, which together

1 implement the described logical functionality of the module. Indeed, a module may
2 include a single instruction, or many instructions, and may even be distributed
3 among several discrete code segments, within different programs, and across
4 several memory devices.

5 Figure 5 illustrates a schematic block diagram of a system 40 for
6 communicating with an Information Management System (IMS) 10 using
7 interchangeable documents according a presently preferred embodiment of the
8 invention. The system 40 may include a Web server 46, which may receive a
9 specially-encoded document 44 from a Web browser 42 via the Internet 48.

10 Various Web servers 46 may be used, such as the Windows NT Server™,
11 available from Microsoft Corporation. Likewise, the Web browser 42 may be
12 implemented using a conventional navigation tool for accessing the World Wide
13 Web (WWW), such as Microsoft Internet Explorer™ or Netscape Navigator™.

14 The document 44 may be encoded to represent one or more IMS transaction
15 definitions 35, such as APPLCTN or TRANSACT macros 36, 38, using an open
16 document format, such as the eXtensible Markup Language (XML). Techniques for
17 encoding IMS transaction definitions 35 in XML are described more fully hereafter.
18 XML is similar to the HyperText Markup Language (HTML), but provides the
19 ability to define custom tags by means of Document Type Definitions (DTDs).

20 In various embodiments, the system 40 also includes an IMS gateway 50,
21 which may provide an interface between the Web server 46 and the IMS 10. The

1 IMS gateway 50 may include an XML to Macro parser 52, which may parse the XML
2 document 44 using a specialized DTD 54 to decode the IMS transaction definition
3 35. A system and method for generating the DTD 54 and the parser 52 is described
4 more fully hereafter. After the transaction definition 35 is decoded, the IMS
5 gateway 50 may provide the same to the IMS 10.

6 In alternative embodiments, the system 40 may be used in reverse to generate
7 an XML document 44 from an IMS transaction definition 35, as illustrated Figure 6.
8 For example, a transaction definition 35 may be obtained from the IMS 10 or another
9 source. Thereafter, a Macro to XML parser 53 within the IMS gateway 50 may use
10 the DTD 54 to encode the transaction definition 35 as an XML document 44. In
11 certain embodiments, two different parsers 52, 53 are used, one for encoding and
12 one for decoding a transaction definition 35, although the invention is not limited
13 in this respect. After the XML document 44 is generated, the Web server 46 may
14 transmit the same to a Web browser 42 or the like via the Internet 48.

15 Figure 7 illustrates a system 60 for generating the DTD 54 and a number of
16 access classes 76 and for creating the parsers 52, 53 based on the DTD 54 and the
17 access classes 76. In various embodiments, the system 60 includes a Uniform
18 Modeling Language (UML) modeling tool 62. The UML modeling tool 62 may be
19 used to produce a UML object model 64 to represent the transaction definitions 35.
20 In one embodiment, the UML modeling tool 62 is Rational Rose™, a visual modeling
21 tool available from Rational Software. Appendix A includes a UML Model Report

1 generated by Rational Rose, as well as a description of various UML classes
2 according to an embodiment of the invention.

3 UML is a language for specifying, visualizing, constructing, and
4 documenting software systems, as well as business models and the like. UML is
5 capable of representing the static, dynamic, environmental, and organizational
6 aspects of almost any system.

7 Figure 8 illustrates a UML object model 64 of the APPLCTN and TRANSACT
8 macros 36, 38, as displayed by the UML modeling tool 62. In one embodiment, the
9 model 64 includes an *ApplicationControlMacro* object 66 to represent the APPLCTN
10 macro 36. Each parameter of the APPLCTN macro 36 may be represented by an
11 attribute 67 of the *ApplicationControlMacro* object 66. For example, the PSBName
12 attribute represents a parameter in the APPLCTN macro 36 for the Program
13 Specification Block (PSB) name.

14 As illustrated, the *ApplicationControlMacro* object 66 may be linked to zero or
15 more *TransactionControlMacro* objects 68. In one embodiment, a
16 *TransactionControlMacro* object 68 represents the TRANSACT macro 38, each
17 attribute 69 of the object 68 corresponding to a parameter of the macro 38.

18 In addition, the UML object model 64 may include a number of enumeration
19 stereotypes 70, which are type definitions of attributes 67, 69 having a fixed set of
20 values. For example, an attribute 67 of the type, *TscheduleType*, may only have a
21 value of "Serial" or "Parallel" in certain embodiments. Likewise, the UML object

Figure 11 illustrates a method 90 for generating an XML document 44 from an IMS transaction definition 35 according to an embodiment of the invention. The

The present invention may be embodied in other specific forms without departing from its scope or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.